

●eenbank Electronics
Telephone: 051-645 3391

ASSEMBLY LANGUAGE AND DEBUG PACKAGE

HiSoft Devpac80

Version 2

Devpac80 The New Standard

Devpac80™ Version 2 is the latest assembly language development tool from HiSoft. It comprises a suite of three major programs running under CP/M for all Amstrad Z80 computers with disc.

ED80 is a powerful, WordStar-compatible, configurable program editor. An essential feature of ED80 is its help screens; although the user can configure all the commands, a single key-press will bring down help screens. These are kept in coded form on the disc so they don't encroach on precious memory space.

The Editor

ED80 features cut-and-paste editing with easy restoration of deleted text. The dynamic block-handling includes delete, move, copy, restore, plus a flexible read and write that can address either the disc, the printer (write only), or even another machine running ED80. There is an advanced "find/substitute" facility with wild-cards and the screen-handling features intelligent side-scrolling with edit-time tab handling. Included in ED80 is the extremely useful "goto <line>" function enabling easy location of assembly-time errors. ED80 also knows all the errors that were produced by the previous assembly and you can step through these one-by-one correcting them with just one keypress. ED80 is only 8K in length and comes with a tutorial and an easy-to-use installation program for tailoring the program to the computer and to the user's preference.

The Assembler

GEN80 is a fast, high-specification 2-pass macro assembler. Handling over 3000 lines of source code per minute, GEN80 features macros, disc inclusion, conditional assembly, production of .REL or .COM files and full maths; Boolean operators, exponents and bit manipulations. The many assembly options include:- generate no code file, generate a symbol table listing to disc/screen/printer, list to the printer, list to the disc, inhibit list, define label length, type in direct from the keyboard at the head of a file and many others for easy control over assembly. Any of the options can appear either on the command line or in the file.

CONDITIONAL ASSEMBLY controls object code generation at assembly-time and as an example allows the user to develop a piece of software for many different machines and then by use of a single label generate each version of the object code from the same source. The feature may also be used during development of a program, where a label may act as a switch controlling the inclusion of sections of debugging code. Nesting of conditionals is supported.

DISC INCLUSION is an extremely powerful feature that allows the assembly of very large files (e.g. 200K of source). The presence in the source-file of the line; *I <filename> will cause GEN80 to search the disc for the named file and then to assemble it as though it were explicitly in the source. This system encourages the modular approach to programming as one may develop and save modules individually (perhaps build up one or more macro library files) and finally assemble a file consisting solely of *I statements. Includes may be nested.

RELOCATABLE OUTPUT (.REL files) allows large programs to be developed in stages and then linked together to produce the final, executable file. The intermediate REL files may also be linked with REL files produced by other languages; e.g. ProPascal™ and ProFORTRAN™, all MicroSoft™ languages, DR CBASIC™ and many more. REL files are considered the standard intermediate format under CP/M, you use the linker LINK.COM supplied with your Amstrad CP/M to combine them together. The ASEG, CSEG and DSEG directives are supported to allow programs to have separate absolute, code and data segments allowing ROMable code to be developed easily. PUBLIC and EXTERNAL symbols allow the passing of labels between programs and the symbol table may be dumped at assembly time or link time for use with the symbolic debugger. External plus offset is supported.

MACROS are handled as detailed in the Zilog specification. Parameters to macros may be either numeric or textual, macro calls may be nested giving a sophisticated recursive capability. Local labels may be declared and used.

Output can be directed to a COM file (for direct execution under CP/M), a REL file (for linking) or a BIN file (for running under AMSDOS) and this can be turned off for a quicker, syntax-check assembly.

The Debugger

MON80 is a comprehensive, highly advanced monitor and debugger. Of course MON80 contains all the normal features such as single-step, dynamic multiple breakpoints and full register and memory display/modify but the sophisticated interpretative execution of MON80 enables features such as conditional breakpoints (e.g. execute until register HL is 0, DE is -1 and the program counter is less than

16384), single-step in ROM, switch memory banks, interruptable code execution (if you suspect your code is crashing at a certain point, you can just let it execute and then interrupt at will), simple profiling with user-defined loop watchpoints (e.g. count how many times you've been through LOOP, optionally with a scale factor). All breakpoint commands handle your program symbols loaded from disc so, for example, you can set a breakpoint at LOOP to check the condition $(de=COUNT/2) .or. ((LINUM)=1000)$.

If true, then a break is made displaying your program state. Very powerful.

The search facilities allow you to find sequences of bytes, strings or even parts of assembler mnemonics. For example you may find the next RLCA instruction or the next instruction involving the BC register pair.

The powerful disassembler allows any number of data areas to be specified and will produce a disk file of any size that can then be assembled using GEN80 or edited using ED80. Symbols from an assembled or linked file may be included for symbolic debugging and these symbols may be modified whilst debugging. You can, of course, symbolically debug any file produced by DR LINK™, MicroSoft L80™ or GEN80 because Devpac80™ supports the standard CP/M linker format.

MON80 comes in two versions, the small version is only 7K allowing very large programs to be debugged whilst the extended version includes the conditional breakpoint code and full symbolic expression handling giving maximum power (it's still less than 12K long!).

Devpac80™ comes with an extensive, ring-bound manual containing a full tutorial to get you started with assembly-language programming.

Order Code: HSDEV8035
Price: 42.95 + VAT